A Goofy Idea for an Exascale File System

August 10, 2011

Lee Ward
Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation

- Current parallel FS technologies all roughly based on the same architecture
  - Notable differences in metadata management
    - But always some centralized form of management & control
  - Utilize storage in much the same way; Striped, static parameters and fixed locations once written
  - Built for POSIX first, seemingly, and high performance second
- It is appropriate to look at the fundamental architecture again
  - Exascale is coming, just don't know when
    - A potential inflection point
    - My user community has said they could tolerate that, this one time
  - Tweaking and bending

Goals

- Storage as a service
  - Leverage LWFS where possible and reasonable
  - Redesign the storage component, entire
- Symmetric
  - Storage servers offer the same API and access to stored data
    - Can provide space or data
    - Alternatively, can help a client locate space or data
- Storage accepts responsibility for data
  - Servers cooperate in order to
    - Achieve resilience guarantees
    - Provide bandwidth where and when needed
- Eliminate, at least mitigate, global state
- Heterogeneous media
  - Type, from DRAM to tape
  - Ages

Membership, Command, and Control

- Heavily P2P inspired protocols
  - Cooperative servers operate as clients when relocating or replicating data
- Membership and status information must be propagated
  - But it's a “sin” to use the network
  - Piggybacked messages?
    - Opportunistic information propagation implies that age should be accounted for in making decisions

Ingest and Update; Choosing a subset

- Client goal is to reasonably maximize use of the NIC and path(s) in the network
  - Lack of global state implies a greedy approach
  - Too greedy (too many servers), though, and variance becomes an issue
- Initial candidates determined from neighbor information
- Refined list obtained from a match between object attributes and server attributes
- Weighted by observed network performance

Some Object Attributes

- Many of the usual, of course; time stamps, permission related, etc.
- Minimum permissible persistence
  - Sufficient authoritative copies must exist at the desired level, or better
- Desired persistence
  - Servers are to achieve sufficient authoritative copies at the desired level, or better
    - Yes, there is API and protocol allowing the protocol to establish that the guarantee has been achieved

Sandia National Laboratories

Some Server Attributes

- Provide information about
  - Capacity, total and used
    - Some idea as to how fast a client might consume space when writing
  - Current and recent load
    - Gauge potential responsiveness
  - Persistence quality
    - Suitability as an initial target
  - Media performance characteristics
    - Latency and bandwidth

Implicit Network Attributes

- Latency, bandwidth, distance
- Provided by low-level network transport

Adapt to the Environment

- An initial choice of subset by the client may not remain optimal
  - Think network failure, cross-traffic, servers unfortunately becoming “hotspots”, low capacity, etc.
  - May not even have been optimal to begin with
    - May learn of better candidates
- We can't change in the middle of a stream!
- Really? Why not?
  - Just need a way to reconcile and determine what is authoritative

Byte-granular, Versioned, Segments

- Let me know when you are done laughing
- Server maintains an “interval” database tracking each update
  - Client may supply a 64-bit version number
  - To be used by both the client and set of servers to reconcile multiple objects
- Performance
  - >10,000 updates/sec
  - >100,000 retrieved segments/sec
- Atomic, coherent, and isolates transactions
  - New version, not yet integrated, is durable
  - But only ~6,000 updates/sec
- Yes, the associated database can outgrow the actual data
- Ok, we may have to admit defeat and move to a block-based system
  - But this gets a fair shot, first!

Migration

- Instantiation or update of an object is unlikely to happen in the final resting place
  - Client probably chose based on a desire for performance
  - Can limit the transient risk by choosing the subset based on advertised persistence, though
- Is even unlikely to have occurred in a "safe" place
  - Desired persistence attribute less than the servers persistence attribute
- But the storage nodes are to assume responsibility
  - The client must cooperate and utilize the supplied protocol

Migration Policy

- Instantiation or update of an object with a desired persistence value greater than the server implies
  - A requirement to instantiate or update a copy on another server or set of servers with “better” persistence
  - Copies and/or erasure codes
- This can be recursive
  - The server is motivated to move the data to a “safer” location
  - Which keeps occurring until sufficient copies are resident on a subset that meets or exceeds the desired persistence

Capacity Management

- Migration will tend to create many redundant copies
- But those nodes must be able to reclaim the space occupied by those copies
- The entire collection of servers functions as a victim cache
  - A server may reclaim the space if it first can determine that the persistence guarantees are sufficient
  - If they are not, it must make them so
- This mechanism does double-duty
  - Reclaim of space by unused copies
  - Capacity balance and rebalance

You Wanted it Back?

- I'm pretty sure it's in there somewhere
  - Unless a critical number of servers have died or gone offline
    - Just one of many open problems
- But where?
  - The system has been allowed to freely move the objects, only constrained by a persistence guarantee

Sandia National Laboratories

Finding Authoritative Copies

- Initial, demonstration, method will be a bounded broadcast
  - Similar to early P2P
- While researching
  - Probabilistic searches that fall back to bounded broadcast
    - Unstructured sensor networks have had good success with this
    - But have issues, requiring shared state in local groups and timely updates
  - A DHT in the lower layers?

Sandia National Laboratories

Achieving Scalable Reads

- Freshly modified objects should offer many copies on multiple storage nodes
  - Yes, there is protocol a client may use to inquire
  - Yes, servers may cache information about what other servers contain
    - But it can become stale
- Older objects or those that migrated quickly to relatively static locations won't
  - Potentially, will need to induce copies on other nodes
  - Probably no single method is correct
    - N:M will need to spread many objects
    - N:1 will need to spread one over a large subset
- Many open problems
  - Many single-client jobs crawling the data can't avoid contention
  - The time to spread copies may be intolerable for large, cooperative jobs

Coherency

- If you must...
- We always require cooperating clients
- For a POSIX interface we could provide local transactions at the servers
  - Normal BEGIN, END/ABORT
- But expect the client(s) to coordinate multiple servers
  - Servers must support PRECOMMIT
  - On which the client may supply their own manager to implement a two-phase protocol
- Alternatively, is our versioned writes support already sufficient?
  - Clients could use a lock manager to control access to segment versions on update
  - Our server could refuse updates if a segment overlaps one or more with a higher version number
  - Again, this requires the clients to cooperate

Sandia National Laboratories

Miscellaneous open questions

- How does one delete an object from this system?
  - It appears that the only way is to stomp every copy in the system, simultaneously
  - Else the thing will just freak out and reinstantiate a “safe” number of copies on a “safe” subset
- How do we tell that an object has become “unsafe”
  - Insufficient copies remain or we need to find a spare for a missing piece involved in an ECC-protected segment

Sandia National Laboratories

Conclusion

- A new approach, the storage collective
  - <Insert Borg joke here>
- Re-examining fundamental design choices
- Storage assumes direct responsibility for resilience and integrity
- Scalable write performance
  - At all sizes, both N:1 and N:M
  - Reads lose, must fix this
- Very much a work-in-progress

Sandia National Laboratories

Thanks!

- To DOE/NNSA and NSF for their continuing support and encouragement
- To the many people who’ve helped make these ideas better (workable)
- To you, for your patience and attention